CLAIMS

1. An apparatus comprising:

at least one processor;

a memory coupled to the at least one processor; and

a database optimizer residing in the memory and executed by the at least one processor, the database optimizer using statistics regarding the type of applications accessing data in a database, the frequency with which the applications access the data, and the location of the data being accessed by the applications to make at least one change to the database schema to optimize the performance of accessing data in the database.

2. The apparatus of claim 1 wherein the database optimizer makes the change to the database schema according to a set of rules that specify a preferred data type for each type of application accessing data in the database.

3. The apparatus of claim 1 wherein the change to the database schema comprises changing the data type of at least one column in the database.

4. The apparatus of claim 1 wherein the change to the database schema comprises adding a new column of a second data type to the database that contains the same data in an existing column of a first data type in the database.

5. The apparatus of claim 4 wherein the database optimizer further comprises a data coherency mechanism for maintaining data coherency between the existing column and the new column.

6. The apparatus of claim 1 wherein the database optimizer receives requests from at least one application to access data in the database, and returns data from the database of a data type that is expected by the requesting application.

7. The apparatus of claim 1 wherein the database optimizer further comprises a run-time statistics gathering mechanism to gather the statistics.

8. The apparatus of claim 1 wherein the database optimizer operates according to customization settings set by a human user.

9. The apparatus of claim 1 wherein the database optimizer further comprises a data type conversion mechanism that converts data in a first data type retrieved from the database to a second data type that is preferred by an application requesting the data.

10. An apparatus comprising:

at least one processor;

a memory coupled to the at least one processor;

a database residing in the memory;

a database optimizer residing in the memory and executed by the at least one processor, the database optimizer comprising:

a data access mechanism that uses statistics regarding the type of applications accessing data in a database, the frequency with which the applications access the data, and the location of the data being accessed by the applications to make at least one change to the database schema to optimize the performance of accessing data in the database;

customization settings that may be set by a human user to determine the function of the database optimizer;

a data coherency mechanism that maintains coherency of reflective columns in the database that are created by the data access mechanism and that contain the same data in different data types; and

a data type conversion mechanism that converts data in a first data type retrieved from the database to a second data type that is preferred by a requesting application.

11. An apparatus comprising:

at least one processor;

a memory coupled to the at least one processor;

a database residing in the memory;

a database optimizer residing in the memory and executed by the at least one processor, the database optimizer comprising:

a mechanism that reads statistics regarding the type of applications accessing data in the database, the frequency with which the applications access the data, and the columns being accessed by the applications;

if the statistics indicate that a selected type of application has a number of accesses to a selected column of a first data type in the database that exceeds a first threshold level, the database optimizer determines whether the statistics indicate that the selected type of application has a number of accesses to the selected column that exceeds a second threshold level, and if so, the database optimizer changes the data type of the selected column in the database;

if the statistics indicate that a selected type of application has a number of accesses to a selected column of a first data type in the database that exceeds a first threshold level, the data optimizer determines whether the statistics indicate that the selected type of application has a number of accesses to the selected column that exceeds a second threshold level, and if not, the data optimizer adds a new column of a second data type to the database that contains the same data in the selected column, the selected column and the new column being defined as reflective columns because they contain the same data in different data types;

wherein the data optimizer detects when one of the plurality of applications requests access to data in the selected column, determines the preferred data type for the requesting application, determines if the data in the selected column is of the preferred data type for the requesting application, and if

(claim 11 continued)

the data in the selected column is of the preferred data type for the requesting application, returning the data in the selected column to the requesting application;

if the data in any column reflective of the selected column is of the preferred data type for the requesting application, the database optimizer returns the data from the reflective column to the requesting application;

if the data in the selected column and in all reflective columns, if any, is not of the preferred data type for the requesting application, the database optimizer converts the data to the preferred data type for the requesting application, and returns the converted data to the requesting application.

12. A method for optimizing a database comprising the steps of:

(1) determining a preferred data type for at least one of a plurality of applications that access the database; and

(2) dynamically changing a schema for the database to provide the preferred data type when at least one of the plurality of applications requests access to data in the database.

13. The method of claim 12 wherein the step of dynamically changing the schema determines what to change according to:

2A) the type of the plurality of applications accessing data in the database;

2B) the frequency with which the plurality of applications access the data; and

2C) the location of the data being accessed by the plurality of applications.

14. The method of claim 12 further comprising the steps of:

(3) determining when one of the plurality of applications accesses the database that has a different preferred data type than the data type specified in the database schema; and

(4) converting the data retrieved from the database to the different preferred data type.

15. The method of claim 12 wherein step of dynamically changing the schema for the database comprises the step of changing the data type of at least one column in the database.

16. The method of claim 12 wherein the step of dynamically changing the schema for the database comprises the step of adding a new column of a second data type to the database that contains the same data in an existing column of a first data type in the database.

17. The method of claim 16 further comprising the step of maintaining data coherency between the existing column and the new column.

18. The method of claim 12 further comprising the step of specifying a preferred data type for at least one of a plurality of applications that access the database.

19. The method of claim 12 further comprising the step of gathering the statistics.

20. A method for reading data from a database comprising the steps of:

(1) specifying a preferred data type for at least one of a plurality of applications that access the database;

(2) detecting when one of the plurality of applications requests access to data in the database;

(3) determining the preferred data type for the requesting application;

(4) determining if the data is stored in the database in the preferred data type for the requesting application;

(5) if the data is stored in the database in the preferred data type for the requesting application, returning the data to the requesting application;

(6) if the data is not stored in the database in the preferred data type for the requesting application, performing the steps of:

(6A) converting the data to the preferred data type for the requesting application; and

(6B) returning the converted data to the requesting application;

(7) reading statistics regarding the type of applications accessing data in the database, the frequency with which the applications access the data, and the location of the data being accessed by the applications; and

(8) dynamically changing a schema for the database to provide the preferred data type when at least one of the plurality of applications requests access to data in the database.

21. The method of claim 20 further comprising the step of gathering the statistics.

22. The method of claim 20 wherein the step of dynamically changing the schema for the database comprises the step of changing the data type of at least one column in the database.

23. The method of claim 20 wherein the step of dynamically changing the schema for the database comprises the step of adding a new column of a second data type to the database that contains the same data in an existing column of a first data type in the database.

24. A method for optimizing accesses to a database comprising the steps of:

(1) reading statistics regarding the type of applications accessing data in the database, the frequency with which the applications access the data, and the columns being accessed by the applications;

(2) if the statistics indicate that a selected type of application has a number of accesses to a selected column of a first data type in the database that exceeds a first threshold level, performing the steps of:

(2A) if the statistics indicate that the selected type of application has a number of accesses to the selected column that exceeds a second threshold level, changing the data type of the selected column in the database;

(2B) if the statistics indicate that the selected type of application has a number of accesses to the selected column that does not exceed a second threshold level, adding a new column of a second data type to the database that contains the same data in the selected column, the selected column and the new column being defined as reflective columns because they contain the same data in different data types.

25. The method of claim 24 wherein the first and second threshold levels may be set by a human user via a user interface.

26. The method of claim 24 further comprising the step of maintaining coherency of data in the selected column and the new column.

27. The method of claim 24 further comprising the step of gathering the statistics.

28. The method of claim 24 further comprising the steps of:

(3) specifying a preferred data type for at least one of a plurality of applications that access the database;

(4) detecting when one of the plurality of applications requests access to data in the selected column;

(5) determining the preferred data type for the requesting application;

(6) determining if the data in the selected column is of the preferred data type for the requesting application;

(7) if the data in the selected column is of the preferred data type for the requesting application, returning the data in the selected column to the requesting application;

(8) determining if the data in any column reflective of the selected column is of the preferred data type for the requesting application;

(9) if the data in a reflective column is of the preferred data type for the requesting application, returning the data from the reflective column to the requesting application;

(10) if the data in the selected column and in all reflective columns, if any, is not of the preferred data type for the requesting application, performing the steps of:

(10A) converting the data to the preferred data type for the requesting application; and

(10B) returning the converted data to the requesting application.

29. A program product comprising:

(A) a database optimizer that uses statistics regarding the type of applications accessing data in a database, the frequency with which the applications access the data, and the location of the data being accessed by the applications to make at least one change to the database schema to optimize the performance of accessing data in the database; and

(B) computer-readable signal bearing media bearing the database optimizer.

30. The program product of claim 29 wherein the computer-readable signal bearing media comprises recordable media.

31. The program product of claim 29 wherein the computer-readable signal bearing media comprises transmission media.

32. The program product of claim 29 wherein the database optimizer makes the change to the database schema according to a set of rules that specify a preferred data type for each type of application accessing data in the database.

33. The program product of claim 29 wherein the change to the database schema comprises changing the data type of at least one column in the database.

34. The program product of claim 29 wherein the change to the database schema comprises adding a new column of a second data type to the database that contains the same data in an existing column of a first data type in the database.

35. The program product of claim 34 wherein the database optimizer further comprises a data coherency mechanism for maintaining coherency between the existing column and the new column.

36. The program product of claim 29 wherein the database optimizer receives requests from at least one application to access data in the database, and returns data from the database of a data type that is expected by the requesting application.

37. The program product of claim 29 wherein the database optimizer further comprises a run-time statistics gathering mechanism to gather the statistics.

38. The program product of claim 29 wherein the database optimizer operates according to customization settings set by a human user.

39. The program product of claim 29 wherein the database optimizer further comprises a data type conversion mechanism that converts data in a first data type retrieved from the database to a second data type that is preferred by an application requesting the data.

40. A program product comprising:

(A) a database optimizer comprising:

a data access mechanism that uses statistics regarding the type of applications accessing data in a database, the frequency with which the applications access the data, and the location of the data being accessed by the applications to make at least one change to the database schema to optimize the performance of accessing data in the database;

customization settings that may be set by a human user to determine the function of the database optimizer;

a data coherency mechanism that maintains coherency of reflective columns in the database that are created by the data access mechanism and that contain the same data in different data types; and

a data type conversion mechanism that converts data in a first data type retrieved from the database to a second data type that is preferred by the requesting application; and

(B) computer-readable signal bearing media bearing the database optimizer.

41. The program product of claim 40 wherein the computer-readable signal bearing media comprises recordable media.

42. The program product of claim 40 wherein the computer-readable signal bearing media comprises transmission media.

43. A program product comprising:

(A) a database optimizer comprising:

a mechanism that reads statistics regarding the type of applications accessing data in the database, the frequency with which the applications access the data, and the columns being accessed by the applications;

if the statistics indicate that a selected type of application has a number of accesses to a selected column of a first data type in the database that exceeds a first threshold level, the database optimizer determines whether the statistics indicate that the selected type of application has a number of accesses to the selected column that exceeds a second threshold level, and if so, the database optimizer changes the data type of the selected column in the database;

if the statistics indicate that a selected type of application has a number of accesses to a selected column of a first data type in the database that exceeds a first threshold level, the data optimizer determines whether the statistics indicate that the selected type of application has a number of accesses to the selected column that exceeds a second threshold level, and if not, the data optimizer adds a new column of a second data type to the database that contains the same data in the selected column, the selected column and the new column being defined as reflective columns because they contain the same data in different data types;

wherein the data optimizer detects when one of the plurality of applications requests access to data in the selected column, determines the preferred data type for the requesting application, determines if the data in the selected column is of the preferred data type for the requesting application, and if the data in the selected column is of the preferred data type for the requesting application, returning the data in the selected column to the requesting application;

(claim 43 continued)

if the data in any column reflective of the selected column is of the preferred data type for the requesting application, the database optimizer returns the data from the reflective column to the requesting application;

if the data in the selected column and in all reflective columns, if any, is not of the preferred data type for the requesting application, the database optimizer converts the data to the preferred data type for the requesting application, and returns the converted data to the requesting application; and

(B) computer-readable signal bearing media bearing the database optimizer.

44. The program product of claim 43 wherein the computer-readable signal bearing media comprises recordable media.

45. The program product of claim 43 wherein the computer-readable signal bearing media comprises transmission media.